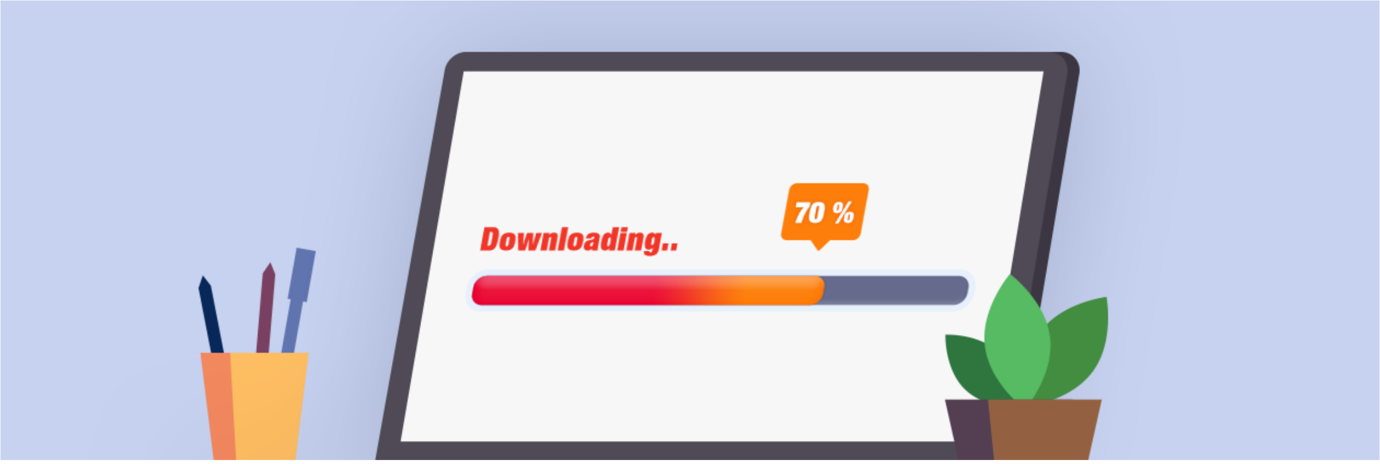
第一章 | 轻松入门，从基础下载方式讲起



热身：最基础的下载方式 —— HTTP(S)

http://dldir1.qq.com/qqfile/qq/TIM2.0.0/22317/TIM2.0.0.exe

这个链接看起来很熟悉是吗？当下载提示弹出来的一刻，你有没有认识到这种最直接最为人熟知也是最多人使用的下载方式，和访问网页这一行为之间的区别？

答案是没有区别。

让我们先来看看 HTTP（全写：HyperText Transfer Protocol）协议的定义：

> HTTP 是一个用于客户端终端（用户）和服务器端（网站）之间请求和应答的协议。

我想你一定不会喜欢文章中充斥着这种晦涩的语言，这是本

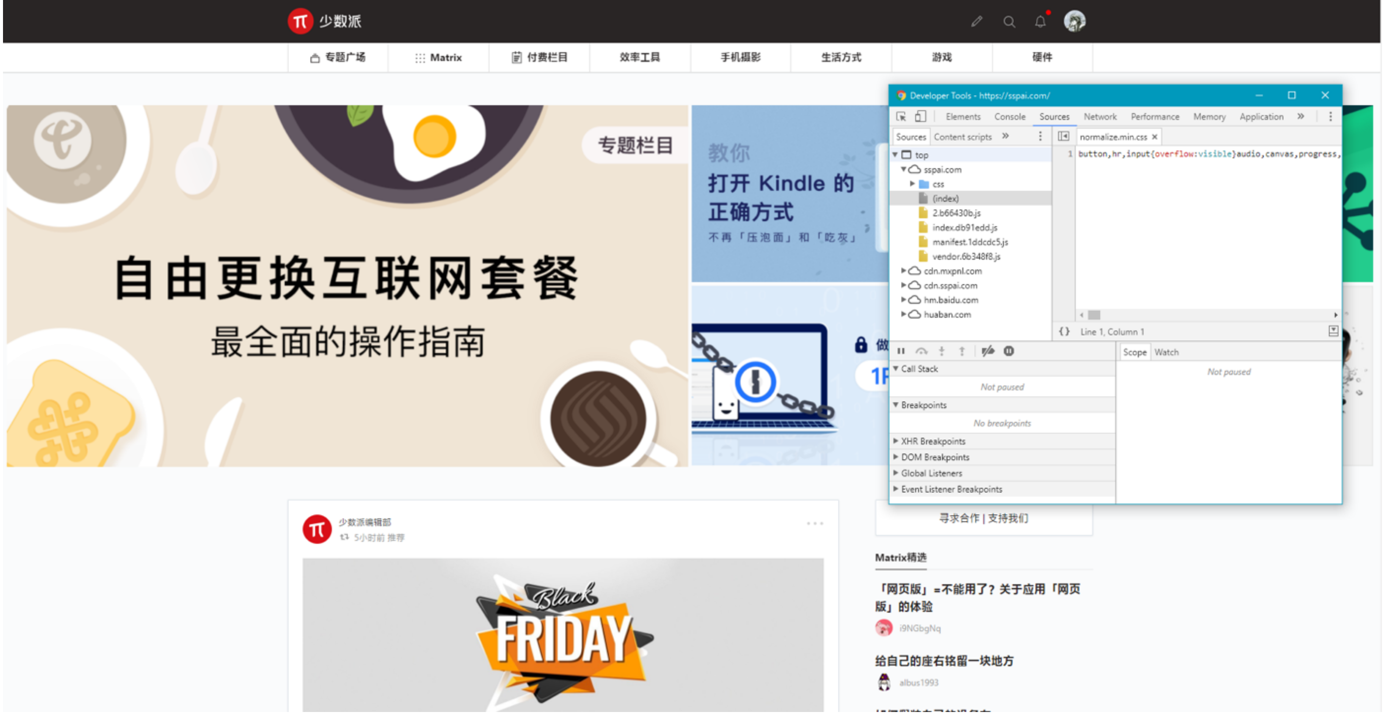
文的解释：

信息即是你在互联网上能访问到的所有东西，包括文本，图像，视频和各种格式的文件，协议是指网络中两台设备之间进行通信所必须共同遵守的规定或规则，HTTP 规定了每次请求的格式，用什么方法传输数据，在各种命令下服务器和浏览器要采取什么响应。

一句话解释：**HTTP 制定了双方该怎么说，用什么方式让对方听到，碰到各种情况如何回应。浏览器和服务器都遵守这个规矩，可以避免出现「鸡同鸭讲」的问题。**

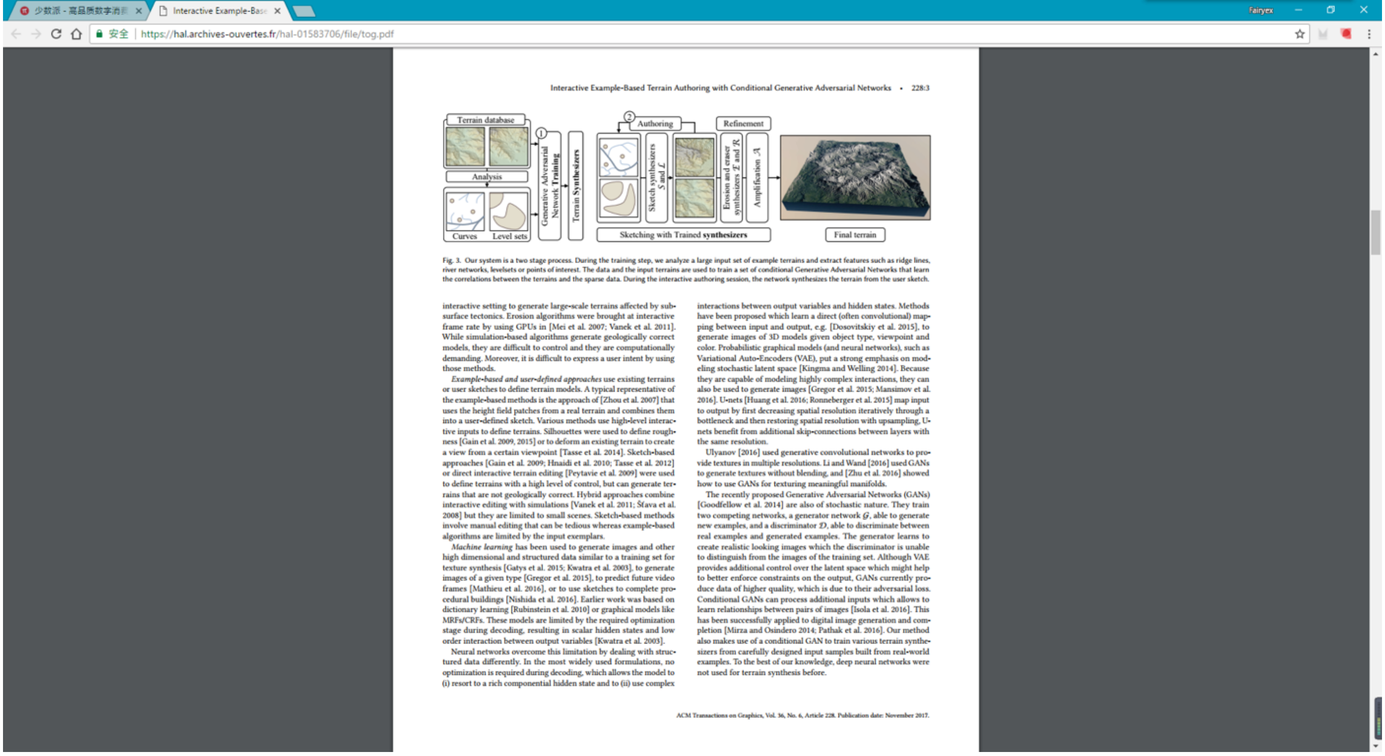
你想要打开少数派首页的时候，浏览器「客户端」会通过对少数派的服务器直接说我想要访问少数派的首页，服务器听到了之后就会动态地生成包括最新文章的首页数据返回给浏览器。

这里的首页数据就是很多个不同格式的文件。之所以我们会常识性地把浏览网页和下载文件区分开，是因为浏览器里面内置了可以将这些数据视觉化的解释引擎「简单理解：解码器」，当首页数据文件下载完毕时立即渲染出可以直接「看」的网页；而一些它「读不懂」的数据，则会提示你保存到设备让其他可以「读懂」的程序打开它。


少数派
专题广场
Matrix
付费栏目
效率工具
手机摄影
生活方式
游戏
硬件
专题栏目
自由更换互联网套餐
最全面的操作指南
打开 Kindle 的
正确方式
Developer Tools - https://sspai.com/
Elements
Console
Sources
Network
Performance
Memory
Application
top
sspai.com
css
(index)
cdn.mxpnl.com
cdn.sspai.com
hm.baidu.com
huaban.com
normalize.min.css
Call Stack
Not paused
Breakpoints
No breakpoints
XHR Breakpoints
DOM Breakpoints
Global Listeners
Event Listener Breakpoints
Scope
Watch
少数派编辑部
Black FRIDAY
寻求合作 | 支持我们
Matrix精选
「网页版」≠不能用了？关于应用「网页版」的体验
给自己的座右铭留一块地方

浏览器中打开控制台（快捷键 Windows：F12；Mac OS：Command + Option + I），在资源（Sources）中能看到浏览器「下载」了哪些文件来显示网页。

举两个实际的例子帮助各位理解：当我们用现代浏览器打开一个 TXT 或者 PDF 文件的时候，往往不会弹出下载框，而是直接在窗口里可以看到内容。原因是现代浏览器内置的解码器比起原始的浏览器多了不少，原本需要下载的 txt 和 pdf 也可以像网页一般浏览了。


少数派 - 高品质数字消费
Interactive Example-Base
安全
https://hal.archives-ouvertes.fr/hal-01583706/file/tog.pdf
Fairyex
Terrain database
Analysis
Curves
Level sets
①
Generative Adversarial Network Training
Terrain Synthesizers
②
Authoring
Sketch synthesizers $\mathcal{S}$ and $\mathcal{L}$
Refinement
Erosion and eraser synthesizers $\mathcal{E}$ and $\mathcal{R}$
Amplification $\mathcal{A}$
Sketching with Trained synthesizers
Final terrain

Interactive Example-Based Terrain Authoring with Conditional Generative Adversarial Networks • 228:3

Fig. 3. Our system is a two stage process. During the training step, we analyze a large input set of example terrains and extract features such as ridge lines, river networks, levelsets or points of interest. The data and the input terrains are used to train a set of conditional Generative Adversarial Networks that learn the correlations between the terrains and the sparse data. During the interactive authoring session, the network synthesizes the terrain from the user sketch.

interactive setting to generate large-scale terrains affected by subsurface tectonics. Erosion algorithms were brought at interactive frame rate by using GPUs in [Mei et al. 2007; Vanek et al. 2011]. While simulation-based algorithms generate geologically correct models, they are difficult to control and they are computationally demanding. Moreover, it is difficult to express a user intent by using those methods.

Example-based and user-defined approaches use existing terrains or user sketches to define terrain models. A typical representative of the example-based methods is the approach of [Zhou et al. 2007] that uses the height field patches from a real terrain and combines them into a user-defined sketch. Various methods use high-level interactive inputs to define terrains. Silhouettes were used to define roughness [Gain et al. 2009, 2015] or to deform an existing terrain to create a view from a certain viewpoint [Tasse et al. 2014]. Sketch-based approaches [Gain et al. 2009; Hnaidi et al. 2010; Tasse et al. 2012] or direct interactive terrain editing [Peytavie et al. 2009] were used to define terrains with a high level of control, but can generate terrains that are not geologically correct. Hybrid approaches combine interactive editing with simulations [Vanek et al. 2011; Šťava et al. 2008] but they are limited to small scenes. Sketch-based methods involve manual editing that can be tedious whereas example-based algorithms are limited by the input exemplars.

Machine learning has been used to generate images and other high dimensional and structured data similar to a training set for texture synthesis [Gatys et al. 2015; Kwatra et al. 2003], to generate images of a given type [Gregor et al. 2015], to predict future video frames [Mathieu et al. 2016], or to use sketches to complete procedural buildings [Nishida et al. 2016]. Earlier work was based on dictionary learning [Rubinstein et al. 2010] or graphical models like MRFs/CRFs. These models are limited by the required optimization stage during decoding, resulting in scalar hidden states and low order interaction between output variables [Kwatra et al. 2003].

Neural networks overcome this limitation by dealing with structured data differently. In the most widely used formulations, no optimization is required during decoding, which allows the model to (i) resort to a rich componential hidden state and to (ii) use complex interactions between output variables and hidden states. Methods have been proposed which learn a direct (often convolutional) mapping between input and output, e.g. [Dosovitskiy et al. 2015], to generate images of 3D models given object type, viewpoint and color. Probabilistic graphical models (and neural networks), such as Variational Auto-Encoders (VAE), put a strong emphasis on modeling stochastic latent space [Kingma and Welling 2014]. Because they are capable of modeling highly complex interactions, they can also be used to generate images [Gregor et al. 2015; Mansimov et al. 2016]. U-nets [Huang et al. 2016; Ronneberger et al. 2015] map input to output by first decreasing spatial resolution iteratively through a bottleneck and then restoring spatial resolution with upsampling. U-nets benefit from additional skip-connections between layers with the same resolution.

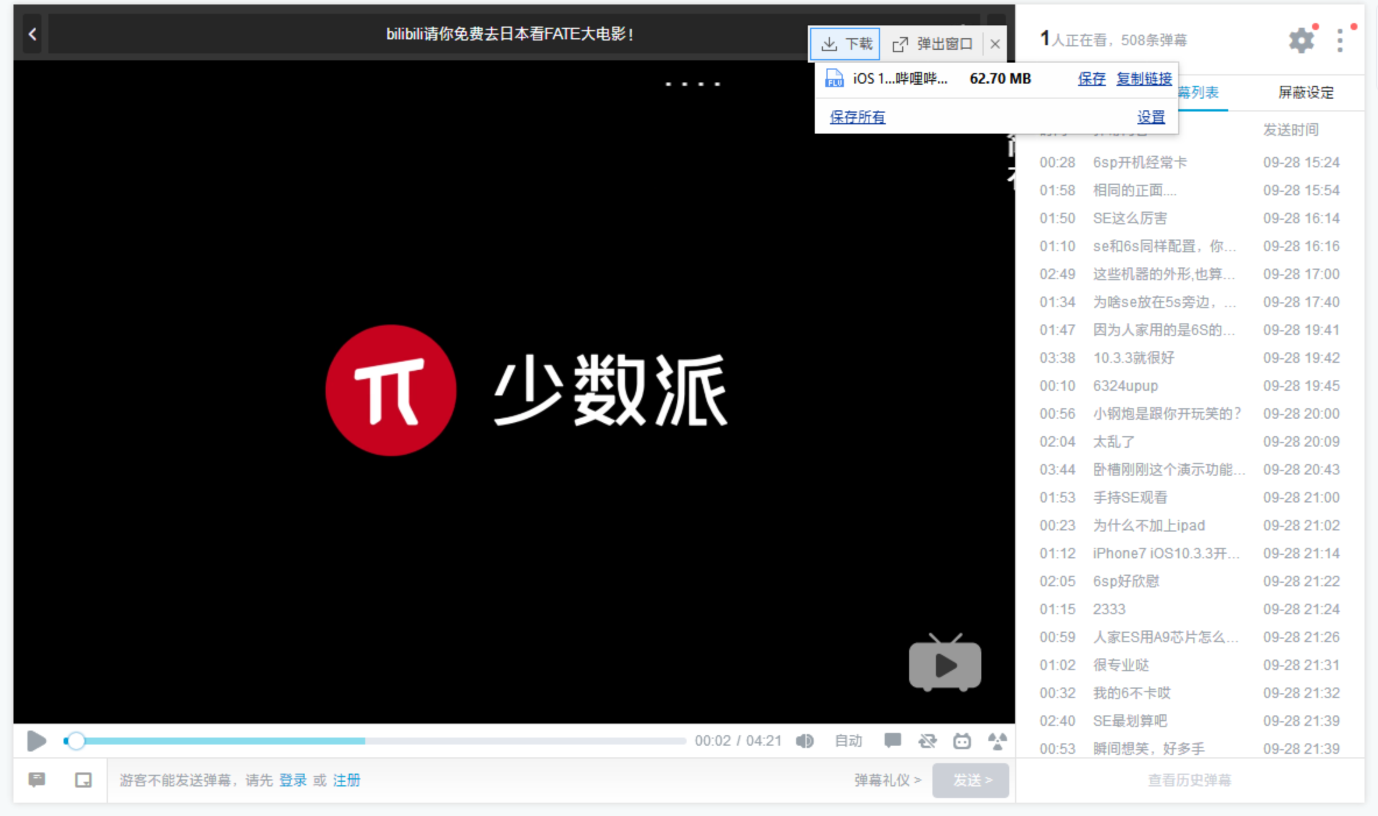
Ulyanov [2016] used generative convolutional networks to provide textures in multiple resolutions. Li and Wand [2016] used GANs to generate textures without blending, and [Zhu et al. 2016] showed how to use GANs for texturing meaningful manifolds.

The recently proposed Generative Adversarial Networks (GANs) [Goodfellow et al. 2014] are also of stochastic nature. They train two competing networks, a generator network $\mathcal{G}$, able to generate new examples, and a discriminator $\mathcal{D}$, able to discriminate between real examples and generated examples. The generator learns to create realistic looking images which the discriminator is unable to distinguish from the images of the training set. Although VAE provides additional control over the latent space which might help to better enforce constraints on the output, GANs currently produce data of higher quality, which is due to their adversarial loss. Conditional GANs can process additional inputs which allows to learn relationships between pairs of images [Isola et al. 2016]. This has been successfully applied to digital image generation and completion [Mirza and Osindero 2014; Pathak et al. 2016]. Our method also makes use of a conditional GAN to train various terrain synthesizers from carefully designed input samples built from real-world examples. To the best of our knowledge, deep neural networks were not used for terrain synthesis before.

ACM Transactions on Graphics, Vol. 36, No. 6, Article 228. Publication date: November 2017.

在 Chrome 中直接浏览 PDF 文件

另一个例子网龄比较长的派友可能还记得，曾经有一种从 IE 缓存目录提取缓存数据然后重命名就可以下载各种在线音频和视频的方法。这种方法侧面证明了浏览互联网内容即是一个不断下载文件的过程。


bilibili请你免费去日本看FATE大电影！
下载
弹出窗口
iOS 1...哔哩哔...
62.70 MB
保存
复制链接
保存所有
设置
1人正在看，508条弹幕
幕列表
屏蔽设定
少数派
00:02 / 04:21
自动
游客不能发送弹幕，请先 登录 或 注册
弹幕礼仪 >
发送 >
查看历史弹幕

		发送时间
00:28	6sp开机经常卡	09-28 15:24
01:58	相同的正面....	09-28 15:54
01:50	SE这么厉害	09-28 16:14
01:10	se和6s同样配置，你...	09-28 16:16
02:49	这些机器的外形,也算...	09-28 17:00
01:34	为啥se放在5s旁边，...	09-28 17:40
01:47	因为人家用的是6S的...	09-28 19:41
03:38	10.3.3就很好	09-28 19:42
00:10	6324upup	09-28 19:45
00:56	小钢炮是跟你开玩笑的？	09-28 20:00
02:04	太乱了	09-28 20:09
03:44	卧槽刚刚这个演示功能...	09-28 20:43
01:53	手持SE观看	09-28 21:00
00:23	为什么不加上ipad	09-28 21:02
01:12	iPhone7 iOS10.3.3开...	09-28 21:14
02:05	6sp好欣慰	09-28 21:22
01:15	2333	09-28 21:24
00:59	人家ES用A9芯片怎么...	09-28 21:26
01:02	很专业哒	09-28 21:31
00:32	我的6不卡哎	09-28 21:32
02:40	SE最划算吧	09-28 21:39
00:53	瞬间想笑，好多手	09-28 21:39

带资源嗅探的浏览器可以轻松下载在线视频

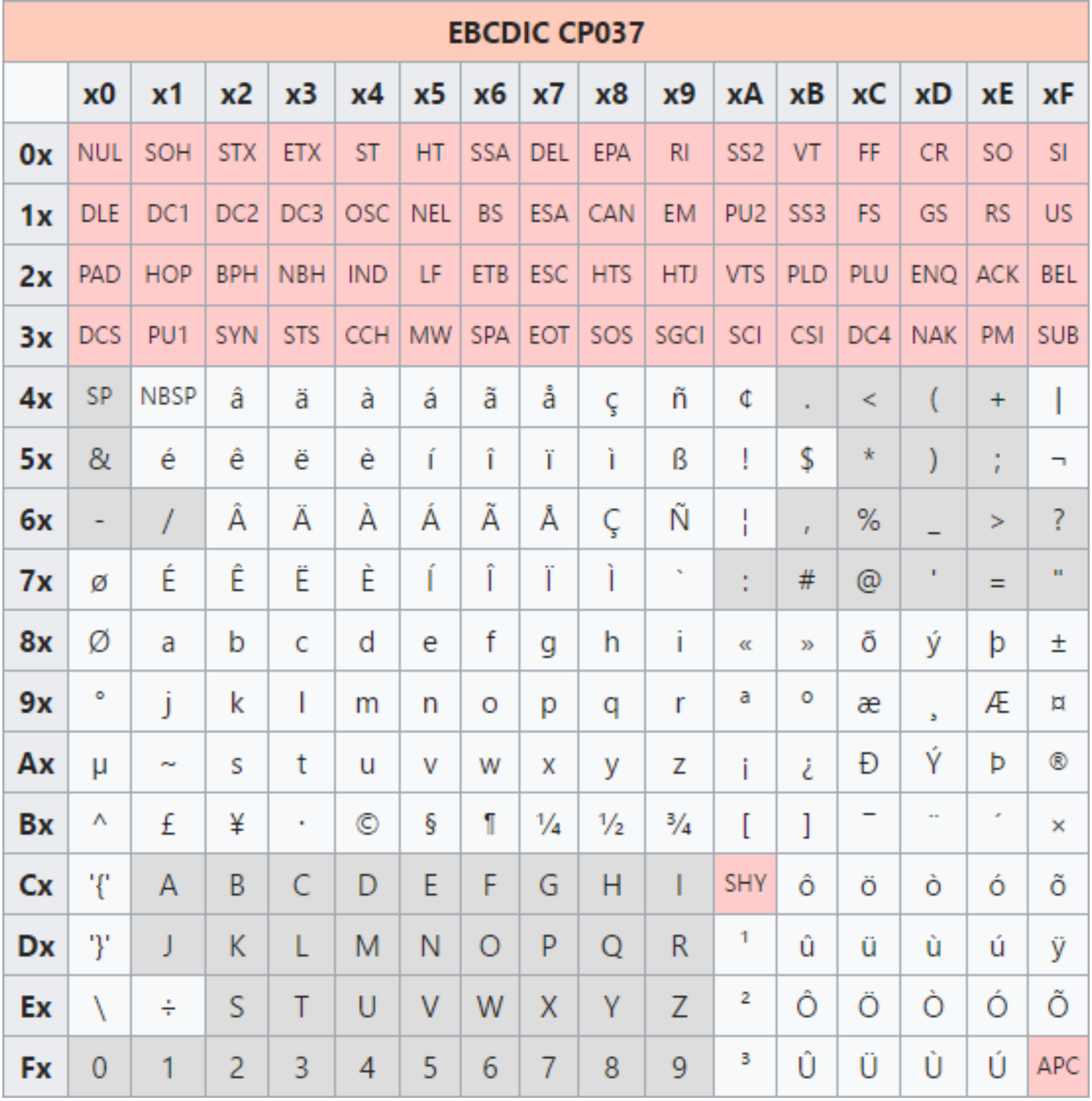
读到这里，大家对这种下载方式应该有个形象的了解了吧。那么我们再来列举一下作为一种传统的下载方式它的优点和缺点。

优点

- **不需要专用的下载工具。**使用普通浏览器甚至不需要浏览器都可以随时随地保存文件。例如我们平时打开需要联网的 App 时获取的数据，就是通过 http 「下载」到本地之后再进行解析。
- 文件名，格式和下载域名直接可见，可以一定程度避免下载到错误的文件。

- **轻松达到满速。**在服务器带宽足够或有好的 CDN 情况下下载速度可以轻松达到物理宽带满速。这里的「带宽」指的是一秒钟能传输的最大信息量，并非实际的速度。单位是 Mbps（Mega bit per second），例如运营商说的 100M（兆）光纤就是 100Mbps 的光纤，注意单位是「**bit（位）**」。我们常说的下载速度是 MB/s（Mega Byte per second），注意单位是「**Byte（字节）**」。虽然缩写相同，但是它们是完全不同的单位，关系是 1Byte=8bit，也就是说 100M 光纤理论上能达到的最大下载速度就是 100/8=12.5MB/s。

那么为什么要用两个单位呢？这是因为数据传输的最小单位是 bit，也就是二进制中的 0 或者 1，而文件储存的最低单位是 Byte，由 8 个 bit 组成，根据排列不同有 2 的 8 次方即 256 种排列方法，可以代表包括数字，大小写字母和常用字符。（这也是为啥 1Byte=8bit，同理一个中文字占 2 byte，因为 2 的 16 次方等于 65536，16bit 才足够覆盖常见中文字符）



EBCDIC CP037

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	NUL	SOH	STX	ETX	ST	HT	SSA	DEL	EPA	RI	SS2	VT	FF	CR	SO	SI
1x	DLE	DC1	DC2	DC3	OSC	NEL	BS	ESA	CAN	EM	PU2	SS3	FS	GS	RS	US
2x	PAD	HOP	BPH	NBH	IND	LF	ETB	ESC	HTS	HTJ	VTS	PLD	PLU	ENQ	ACK	BEL
3x	DCS	PU1	SYN	STS	CCH	MW	SPA	EOT	SOS	SGCI	SCI	CSI	DC4	NAK	PM	SUB
4x	SP	NBSP	â	ä	à	á	ã	å	ç	ñ	¢	.	<	(	+	|
5x	&	é	ê	ë	è	í	î	ï	ì	ß	!	$	*	)	;	¬
6x	-	/	Â	Ä	À	Á	Ã	Å	Ç	Ñ	¦	,	%	_	>	?
7x	ø	É	Ê	Ë	È	Í	Î	Ï	Ì	`	:	#	@	'	=	"
8x	Ø	a	b	c	d	e	f	g	h	i	«	»	ð	ý	þ	±
9x	°	j	k	l	m	n	o	p	q	r	ª	º	æ	¸	Æ	¤
Ax	µ	~	s	t	u	v	w	x	y	z	¡	¿	Ð	Ý	Þ	®
Bx	^	£	¥	·	©	§	¶	¼	½	¾	[	]	¯	¨	´	×
Cx	'{'	A	B	C	D	E	F	G	H	I	SHY	ô	ö	ò	ó	õ
Dx	'}'	J	K	L	M	N	O	P	Q	R	¹	û	ü	ù	ú	ÿ
Ex	\	÷	S	T	U	V	W	X	Y	Z	²	Ô	Ö	Ò	Ó	Õ
Fx	0	1	2	3	4	5	6	7	8	9	³	Û	Ü	Ù	Ú	APC

IBM 最早的电脑上采用的 EBCDIC 编码表就是 8bit，很多人认为这也是 1Byte=8bit 的重要原因

所以数据传输领域经常用 bit 做单位（传输的是 0/1），下载和上传速度用 Byte 做单位（因为下载或上传 1byte 以上的数据才有意义）。

CDN（Content Delivery Network，内容分发网络），是为了

解决网络延时，响应速度和稳定率的一种技术。假设某个网站只有在上海有服务器，那么上海的用户访问网站就比需要经过多个运营商节点跳转的云南用户快上不少。

这时候网站购买了 CDN 服务，在空闲时将网站访问量大的数据复制到覆盖全国的多个服务器上，用户访问网站时自动调用最近的 CDN 服务器上的网站数据，不会再因为距离过远导致数据加载速度降低。既提高了用户体验，又实现了负载均衡和数据备份，现在大部分网站和游戏都部署了 CDN。

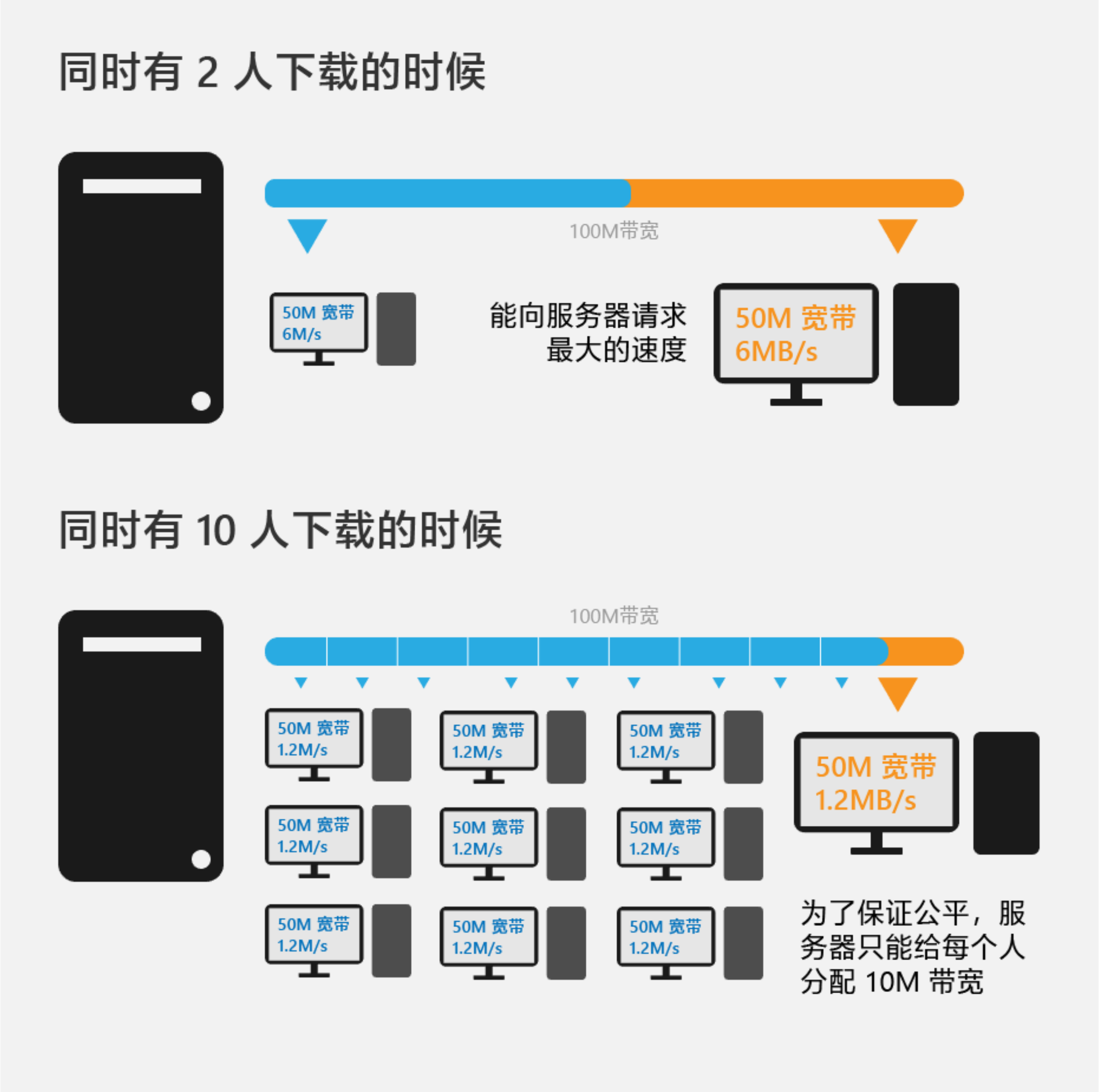
缺点

- 文件是否能被下载完全取决于源地址的所有者，如果请求的文件从源地址删除，移动或重命名，就没有办法下载到。

- 开放式请求，服务器会记录下载者的 IP 地址和其他数据，易造成隐私泄露。

- 下载时文件数据易被中间人篡改，文件数据不安全。由于 HTTP 传输文件时是直接明文传输数据，所以黑客很容易通过监听你的设备流量轻松提取下载的文件，就像边听别人的谈话边抄下来一样。同时黑客还可以充当下载双方的「中间人」，随意拦截修改这些明文数据，伪造正常的请求返回给双方，让服务器和客户端都认为是在和对方通信，让下载方不知道这些文件其实是被篡改过的。

- 在固定服务器带宽下，下载人数越多速度越慢，并且大多数服务器会主动限速。大多数情况使用多线程下载工具可以解决问题。


同时有 2 人下载的时候
100M带宽
50M 宽带
6M/s
能向服务器请求
最大的速度
50M 宽带
6MB/s
同时有 10 人下载的时候
100M带宽
50M 宽带
1.2M/s
50M 宽带
1.2MB/s
为了保证公平，服务器只能给每个人分配 10M 带宽

提示： 只通过采用 https（全写：Hyper Text Transfer Protocol over Secure Socket Layer；简单理解：HTTP 的安全版）的网站下载文件的方法可以一定程度上保证文件数据的安全。具体可以查看本教程第七章关于下载安全问题的部

分。

断点续传与多线程下载加速的原理

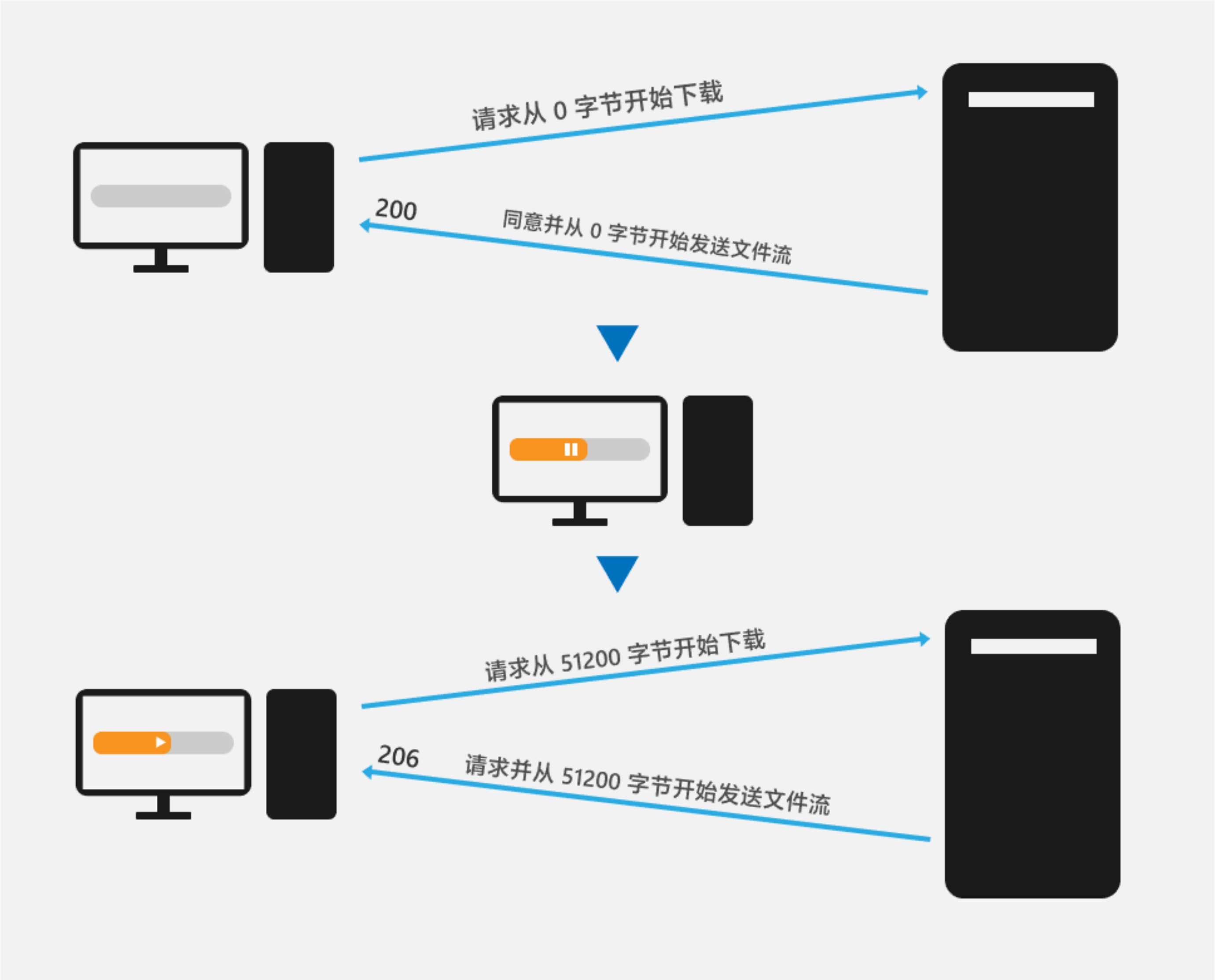
说到 HTTP 下载，那不得不提到一些现在或者曾经非常好用的 HTTP 下载器，比如 Windows 上老牌的 IDM（Internet Download Manager），活在人们心中的网络蚂蚁和网际快车，被誉为手机版 IDM 的 ADM（Advanced Download Manager）等。它们之所以受到人们的喜爱与称赞，是因为它们提供了曾经是杀手锏级别的两个功能 —— 断点续传与多线程下载，这两个功能节省了大家巨量的下载时间，但是你知道这两个功能是怎样工作的吗?

断点续传的原理

断点续传的原理其实非常简单。我们平时下载的文件，无论是 MP3，FLV 还是其他随便什么格式，服务器都是采用流（Stream）的方式传到我们的本地设备上。就像往杯里倒水一样，不管是橙汁还是牛奶都是一样地从瓶里流到杯里，带宽即相当于瓶口的宽度，决定你能倒多快。

以一个 1MB 的文件为例，当你下载了 512KB 然后按下暂停下载按钮的时候，什么也不会发生。只有在你下次继续下载的时候，支持断点续传的客户端才会以字节数的方式直接在请求头（Header，每次请求就像寄信，Header 就像是信封上的信息，包含双方地址身份信息和其他自定义信息）告诉服务器我需要从第 512000 字节开始下载，支持断点下载的服务器（现在不支持这个的服务器已经濒临灭绝了……）就

会从 512000 字节开始传输流并返回一个表示「这是续传」的特殊状态码 206（状态码用来表示目标服务器或者域名解析服务器对这个请求的「态度」，像是我们常见的 404 就是表示「我不存在」的状态码）以说明这个不是重新下载的流。


请求从 0 字节开始下载
200
同意并从 0 字节开始发送文件流
请求从 51200 字节开始下载
206
请求并从 51200 字节开始发送文件流

到这里断点续传还要解决一个问题，就是当暂停下载文件之后，服务器上的文件被修改了，如果接着下载的话最后文件数据肯定是错的。所以支持断点续传的客户端在请求续传时，除了告诉服务器已经下载的大小，还会告诉服务器这个文件在上次暂停时最后修改的时间（服务器文件在这之后修改过的话时间就对不上了）或者整个文件的文件指纹（由文

件大小和内容计算得出，就像指纹一样，内容不同的文件不会有一样的指纹）。

服务器比较之后决定是续传还是从头开始传输文件流给客户端，为了避免服务器端重新传输文件流而客户端认为是续传的文件流，从头开始下载的时候服务器会返回另一个表示「我很正常」状态码（200）。用 200、206 两个状态码即可保证客户端和服务器不会误解对方。

多线程下载加速的原理

多线程下载其实是利用断点续传的一个机智技巧，在说明为什么之前我们来做道简单的数学题（为方便理解设定 1M 带宽等于 1M/s 下载速度，实际下载速度应为 1/8 左右，同时所有人都有比下载速度高的带宽）。

一般的服务器（除去那些故意限制用户速度的）是通过同时下载的线程数平均分配下载带宽，假设一个服务器上的某个文件最高享有 100M 的带宽，同时有 50 个线程下载的时候每个线程分到的下载速度即是 2m/s。可是单纯的服务器认为一个人只能用一个线程来下载，怎想到其中 10 个线程都是同一个人用多线程下载器请求的，那这个人分到了多少下载速度?

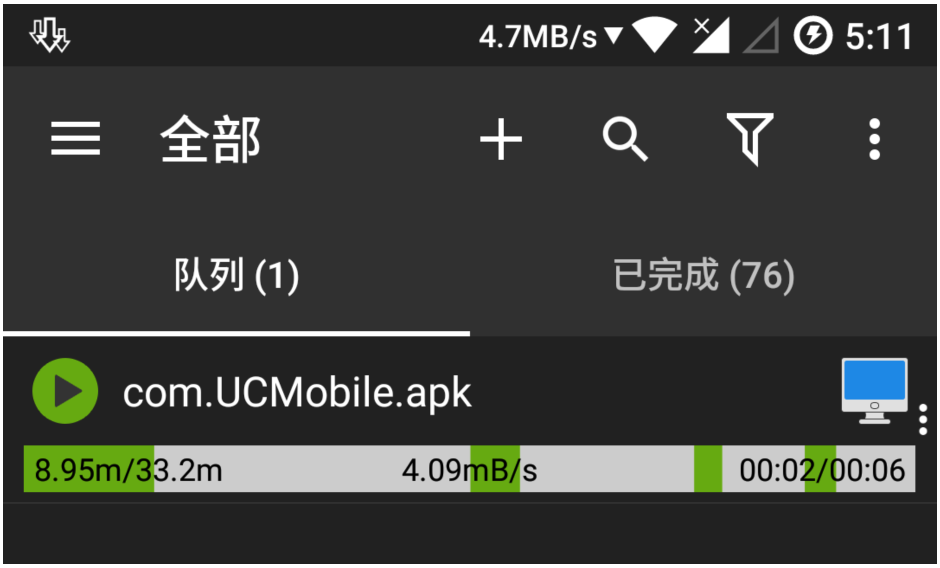
20M/s！知道多线程下载为什么能够提升下载速度了吧，那么多线程实际上是怎样「欺骗」服务器的呢？答案是利用断点续传，简单来讲，一个 10 线程的下载器在接到下载一个文件的请求之后：

- 首先做的就是向服务器请求这个文件的大小然后切成 10 份。

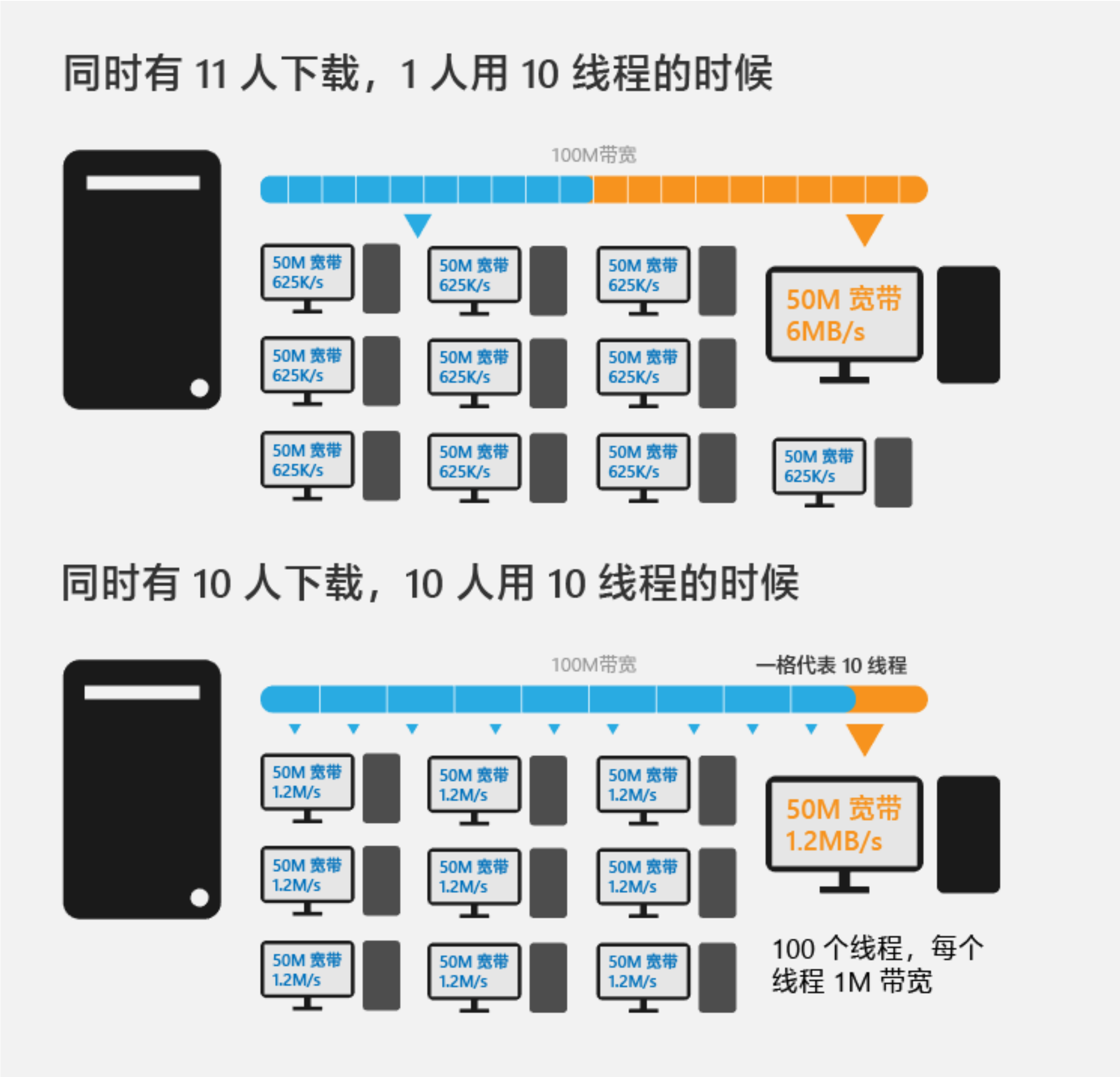
- 接着每一个线程假装断点续传，分别在请求下载的请求头中告诉服务器我要从 1/10，2/10，3/10...9/10 开始下载，服务器天真地以为这真的是一个断点续传的请求于是便从这个线程要求的地方开始传输文件流。

- 只需要 1/10 时间（完美情况下）文件已经下载完毕，下载器再在本地把这十块文件拼起来，一次机智的多线程下载就这样子完成啦。

多线程另外一个可以加快下载速度的原因是 TCP（Transmission Control Protocol 传输控制协议，HTTP 客户端与服务器传输数据的方法）设计之初非常为他人着想，能用少带宽办到的事情绝对不会申请更多的带宽，尽量留给其他人更多的带宽。这对于互联网环境当然是好事，但对于下载者来说「舍己为人」可不行，于是人们开始利用多线程打破这个限制。


4.7MB/s
5:11
全部
队列 (1)
已完成 (76)
com.UCMobile.apk
8.95m/33.2m
4.09mB/s
00:02/00:06

虽然多线程下载听起来很美好，但随着时间的推移，多线程下载的好处正变得越来越不明显。过去大家都在用单线程下载的时代，你用一个网络蚂蚁或者网际快车就能抢到能跑满宽带的带宽。现在大家都开始使用多线程下载器，甚至大部分浏览器都自带了多线程下载，比如前面的例子里面剩下的 49 个人也开始使用 10 线程的下载器，就算你仍然用多线程下载，分到的速度还是会变成 100M/500*10=2M/s，你再也不能比别人快上几倍了。



同时有 11 人下载，1 人用 10 线程的时候

100M带宽
50M 宽带
625K/s
50M 宽带
6MB/s

同时有 10 人下载，10 人用 10 线程的时候

100M带宽
一格代表 10 线程
50M 宽带
1.2M/s
50M 宽带
1.2MB/s

100 个线程，每个线程 1M 带宽

那设置更多的线程数能解决问题吗？能但是效果并不明显，因为传输和线程与服务器请求之间的损耗，除了加重服务器和自己设备内存的压力之外加速效果并不明显，得不偿失。另外，站长们也不是傻子，现在大多数服务器都加上了同一 IP 的线程限制，除非使用类似爬虫一般的代理下载不然达到限制之后再开多少线程都没用。按照目前的情况来讲 20–30 线程能达到一个比较好的加速效果，不过线程数也会「通货膨胀」，大家的确有需要的话再提高多线程下载器的线程哦。

专为传输文件的下载方式 —— FTP

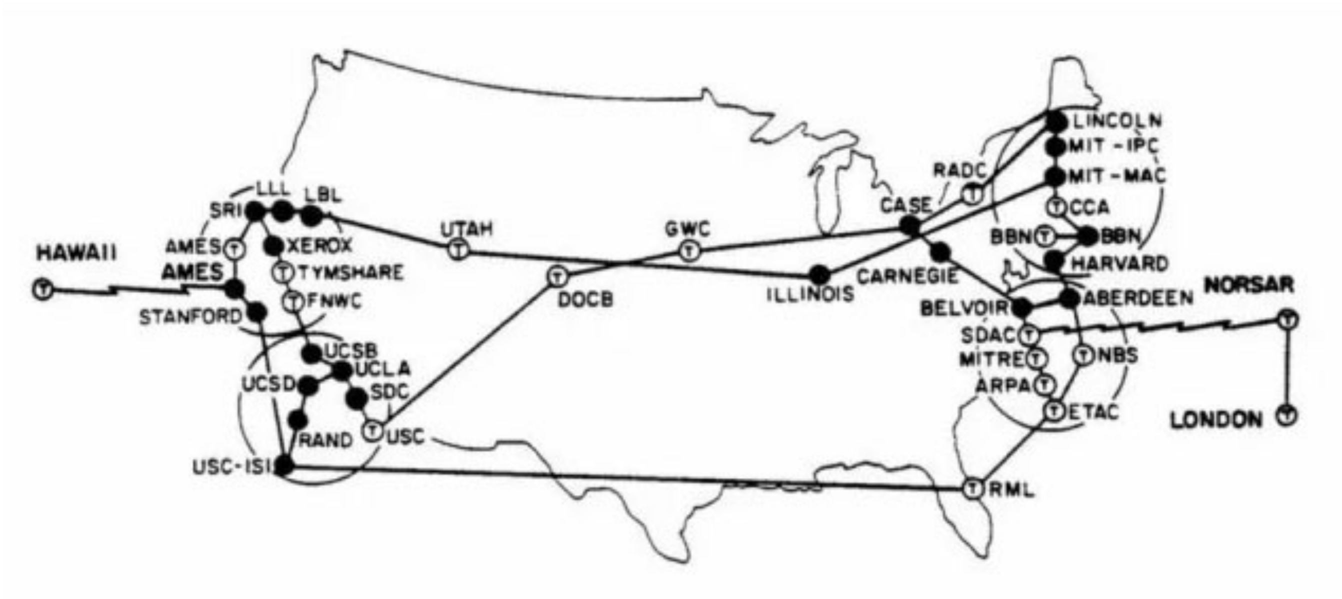
ftp:// 开头的链接相信大家都见过，也下载过，感觉上和 HTTP 下载没有什么两样。那么你知道它和 HTTP 的区别吗?

要说他俩，那就要详细说一下前面提到的它俩的母亲 —— TCP 协议。

TCP 协议和我们熟悉的 IP 一起组成的 TCP/IP 协议族是互联网的基础，TCP 定义了数据如何传输，IP 定义了如何联系互联网上的设备。HTTP，FTP 和其他协议都是从它们身上延伸出来的，可以说 HTTP 和 FTP 是亲兄弟。

不过大哥可不是 HTTP。在 1991 年年轻的 HTTP0.9 初出茅庐然后华丽丽地被无视之前，FTP 已经活跃了十几年，事实上 FTP（1971） 比 HTTP（1989） 的诞生早了 18 年左右，甚至比现在定义的互联网出现的还要早！

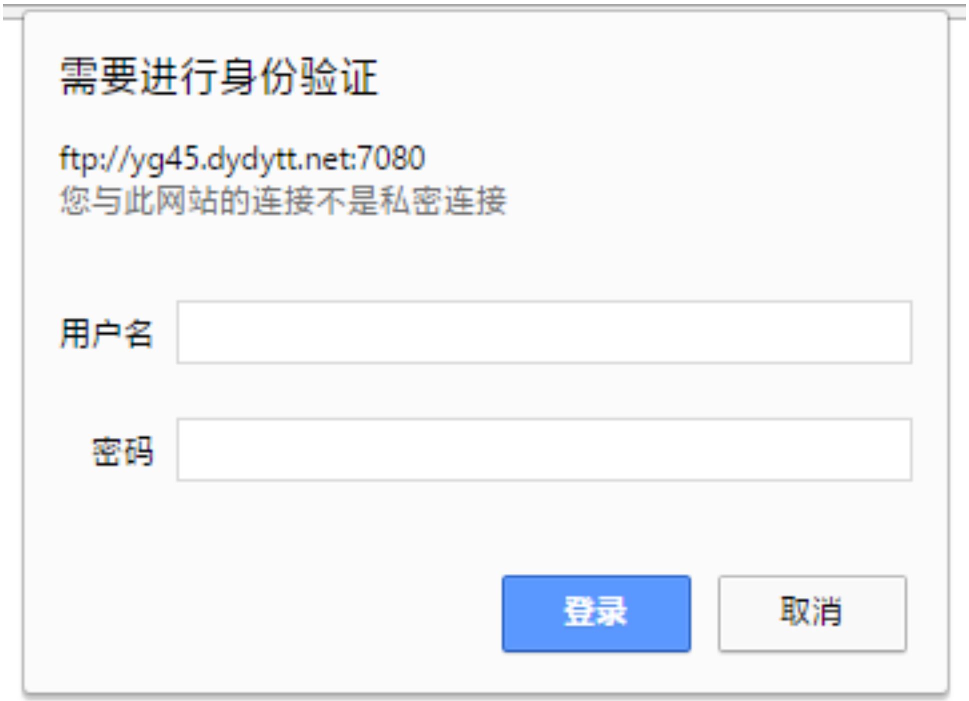
大家知道互联网并不是世界上最早的网络吗？互联网的祖先—— ARPAnet（Advanced Research Projects Agency Network，高等研究计划署网络，音译阿帕网）才是真正的始祖。这个网络采用的通信协议是 NCP（Network Control Program，网络控制程式），那个时候 TCP/IP 和 HTTP 还没影，基于 NCP 的 FTP 就已经出现了，作为阿帕网中计算机之间传输文件的协议。后来阿帕网逐渐变成互联网，NCP 也变成了 TCP/IP，FTP 才把底层协议换成 TCP。所以说，FTP 才是当之无愧的大哥。


HAWAII
AMES
AMES
SRI
LLL
LBL
XEROX
TYMSHARE
FNWC
STANFORD
UCSB
UCLA
UCSD
SDC
RAND
USC
USC-ISI
UTAH
DOCB
GWC
ILLINOIS
CASE
RADC
CARNEGIE
LINCOLN
MIT-IPC
MIT-MAC
CCA
BBN
BBN
HARVARD
ABERDEEN
BELVOIR
NORSAR
SDAC
NBS
MITRE
ARPA
ETAC
LONDON
RML

最初的阿帕网只有 4 个节点，比现在一个宿舍的节点还少

虽然 HTTP 和 FTP 都是同一个协议拓展出来的，不过亲兄弟不一定是双胞胎。HTTP 多是针对为客户端提供文件传输；FTP 则是为了在特定主机之间相互传输文件而开发的标准（互联网出现之前还没有浏览器这种客户端）。

由于涉及到双向传输数据，因此在连接 FTP 服务器的时候我们必须通过用户名和密码认证自己的身份。


需要进行身份验证

ftp://yg45.dydytt.net:7080
您与此网站的连接不是私密连接

那为什么我们通过 FTP 下载的时候不用输入用户名和密码呢？因为密码是可选的，只要服务器以 Anonymous（中文：匿名）作为用户名，那么它储存的文件任何人都可以访问。

除了结构上有所不同以外，在下载时 HTTP 和 FTP 的方便性和下载时间是一模一样的，这意味着它们的优缺点也差不多。FTP 主要是一些需要在服务器上同时储存和交换大量文件的站点在使用，像是大名鼎鼎的电影天堂就以 FTP 作为下载他们电影的唯一方式（虚构链接，仅作展示）：

ftp://ygdy8:ygdy8@yg45.dydytt.net:7080/[****][English][1080p]The Story of SSpai.mkv

附上 FTP 链接的结构解释：

ftp：//[用户名：口令@]ftp服务器域名或 IP：[端口号]/文件路径/文件名。

另外，在很多部电影特别是特工电影中都出现过 FTP 的身影，比如 Netflix 2017 年的电影《王牌保镖》（Hitman's Bodyguard）第 98 分钟就出现了登录 FTP 输入密码的全过程。感兴趣的朋友可以看下这部电影。

你刚才所说的话是否有证据证明
有
我上传到了FTP网站 搜索雷文257
需要输入密码
不好意思 密码是 杜霍维奇是混蛋
除了混蛋小写以外 全都大写

《王牌保镖》片段

匿名 FTP 的特性还受到各个字幕组和游戏破解组的青睐，像是大名鼎鼎的游戏破解组 CPY 就是通过 FTP 发布新的破解游戏。

下节预告

这一节我们利用讲解互联网的基础下载方式来给大家热了热身。如果你觉得这节的货不够「干」，那么请准备好水和笔，因为下一节我们会正式走进现代最先进的下载方式，深入探讨各种各样的 P2P 下载原理（BT，PT 下载，eD2k，磁力，Metalink），绝对不容错过。